Combining Hierarchical Structures on Graphs and Normalized Cut for Image Segmentation

Marco Antonio Garcia Carvalho and André Luis Costa
School of Technology, University of Campinas
Brazil

1. Introduction

The image segmentation task is to divide an image into regions of interest that are suitable for machine or human operations. The machine ability of recognizing and distinguishing objects, or its parts in a scene, is the main goal of computer vision domain. This is a critical issue because the judgement of good or bad segmentation is usually subject to humans.

Extensive studies have been accomplished for image segmentation. The segmentation algorithms are commonly categorized according to the image characteristics borders and regions (Gonzales & Woods, 2000). In the first one, the image is divided based on its discontinuities, *i.e.*, the places where abrupt intensity changes occur. Regarding to the region segmentation, this happen when there are similarities of color or texture, for example, between neighboring pixels. In spite of these categories, the problem is to find a good partitioning of an image among several possible to be achieved.

Several algorithms in image segmentation can be formulated from the partitioning of graphs. This means using graphs as image models or representations and then apply a criterion or methodology in order to split it into subgraphs. The existing literature on graph partitioning is wide, but we are interested on a particular approach, the called Normalized Cut introduced by Shi & Malik (1997).

A graph cut consists of removing edges consistently in order to generate two subgraphs. The Normalized Cut approach is a graph-cut technique based on Spectral Graph Theory responsible for generating balanced subgraphs through the removal of the smallest possible number of edges. The Normalized Cut approach uses concepts by Fiedler (1975) in the manipulation of the second smallest eigenvector of the graph representative matrix as a guide for graph partitioning. The inherent bias of this technique is that balanced partitions for image segmentation cannot be appropriate for some images when small number of partitions is desired (*e.g.*, images with an easily detectible object and an uniform background). A survey of application of Spectral Graph Theory is given by Spielman (2007).

There are several ways of generating the graph model representing the input images. The graph commonly used as input to the Normalized Cut implementation is that one based on the pixel grid similarity. We propose two alternative representations of graphs known as Quadtree and Component Tree similarity graphs. These graphs decompose the image into partitions and thus carry hierarchical information that can be useful on segmentation task.

Another goal is to reduce the computational cost, due to the reduction of the graph size, compared to using the pixel similarity graph.

Finally, we show experiments using Normalized Cut and Quadtree and Component Tree similarity graphs. In addition to the Component Tree, we propose the use of the Reverse Component Tree in order to profit the relevant information in decomposition process of an image. The results are classified using a benchmark provide by The Berkeley Image Database (Martin et al., 2001).

This chapter is organized as follows. Section 2 introduces graph concepts, including our Quadtree and Component Tree based similarity graph. Section 3 presents graph cuts and Normalized Cut theory. Some related works also are described in this section. An overview of the proposed approach is given in Section 4. Experiments in sampled images are done in Section 5 and further comments of the experiments, conclusions, as well as suggestions of future work are done in Section 6.

2. Graph representation

In digital image processing, a graph is commonly used to model digital images (Wilson & Watkins, 1990). In the Normalized Cut segmentation technique the input graph is called *Similarity Graph* (Shi & Malik, 2000), that we explain in details in Section 3.3. In a similarity graph the edge weights should reflect the similarity between the nodes connected by them, and are given by a *Similarity Function*. Here we present the methods for building a similarity graph that we have used in the experiments described in Section 5.

There are several approaches to represent an image as a similarity graph. In the next subsections we present some fundamental concepts on graphs, and four image-graph representation approaches: based on the *Pixel Grid* (Shi & Malik, 2000) , *Multiscale Pixel Grid* (Cour et al., 2005), based on the *Component Tree* (Carvalho, Costa, Ferreira & Cesar-Jr., 2010), and based on the *Quadtree* (Carvalho, Costa & Ferreira, 2010). Approaches based on the *Component Tree*, *Quadtree*, and *Multiscale Pixel Grid* rely on hierarchical structures to model an image as a graph, and provide different segmentation results when compared to the classical non-hierarchical pixel grid approach.

2.1 Basic concepts on graphs

A *graph* is a mathematical structure employed to model or to describe objects and their relationships, *e. g.*, a composition relationship can describe objects and their constituent parts. Let $G = (V, E, W)$ be a non-directed weighted graph; V is a set of *nodes*, E is a set of *edges* $e(i, j)$, $i, j \in V$, and W is a set of *weights* $w(i, j)$, $i, j \in V$. For each edge $e(i, j) \in E$ exists an associated weight $w(i, j) \in W$, that can be represented by a single value or a set of values. Two nodes i and j are *adjacent*, represented by $i \sim j$, if there is an edge connecting i and j. Given a node $i \in V$, its *degree* d_i corresponds to its number of neighbours, and its *strength* s_i (Wilson & Watkins, 1990) to the sum of its edge's weights. A *subgraph* of a graph $G = (V, E, W)$ is a graph $G' = (V', E', W')$ where $V' \subseteq V$, $E' \subseteq E$, and $W' \subseteq W$.

A *path* $\pi = (i_1, i_2, \ldots, i_n)$, $i_n \in V$, is a sequence without repeated nodes where $i_k \sim i_{k+1}$, $k = 1, 2, \ldots, n-1$. Two nodes are *connected* if there exists at least one path between i and j. In a *connected graph* G all pair of nodes are connected. A *cycle* is a path where $i_1 = i_n$. A *tree* is a connected graph with no cycles.

2.1.1 Graphs and matrices

A graph and its features can be represented using matrices (Wilson & Watkins, 1990). The adjacency between graph nodes can be described by an *Adjacency Matrix* A with size $n \times n$, where n is the number of nodes, $|V|$, of graph $G = (V, E, W)$. The matrix elements $a(i, j)$ are 1 if $i \sim j$, $i, j \in V$, or 0, otherwise. Similarly, the *Weight Matrix* W, also with size $n \times n$, $n = |V|$, can store the graph weights $w(i, j)$, $i \sim j$, $i, j \in V$. The *Degree Matrix* D is diagonal with $d(i, i) = d_i$, where d_i is the degree os node $i \in V$. Finally, the *Laplacian Matrix* L is defined as $L = D - A$ for unweighted graphs, or $L = D - W$ for weighted graphs. The Laplacian matrix is commonly used on Spectral Graph Theory (Spielman, 2007).

2.2 Graph based on the Pixel Grid

In this *Pixel Grid-based* image-graph representation each pixel is taken as a graph node, and two pixels within a r distance are connected by an edge. Shi & Malik (2000) have used this approach as the first one for their Normalized Cut technique. This approach have been introduced in the experiments as a landmark for compare results with other approaches.

2.2.1 Multiscale Pixel Grid

The *Multiscale Pixel Grid* graph decomposition algorithm introduced by Cour et al. (2005) works on multiple scales of the image grid to capture coarse and fine detail levels. The construction of the similarity graph is done according to their spatial separation, as in the following Equation

$$W = W_1 + W_2 + \ldots + W_s, \tag{1}$$

where W is a weight matrix that represents a graph composed by independent subgraphs W_s, s corresponds to a scaled pixel grid.

In the Multiscale approach there exists one different radius for each image scale s. Thus, two pixels i and $j \in W_s$ are connected only if the distance between them is lower than a radius R_s. The radii values are a tradeoff between the computation cost and the segmentation result. The Multiscale approach can alleviate this situation by using recursive sub-sampling of the image pixel grid.

2.3 Graph based on the Component Tree

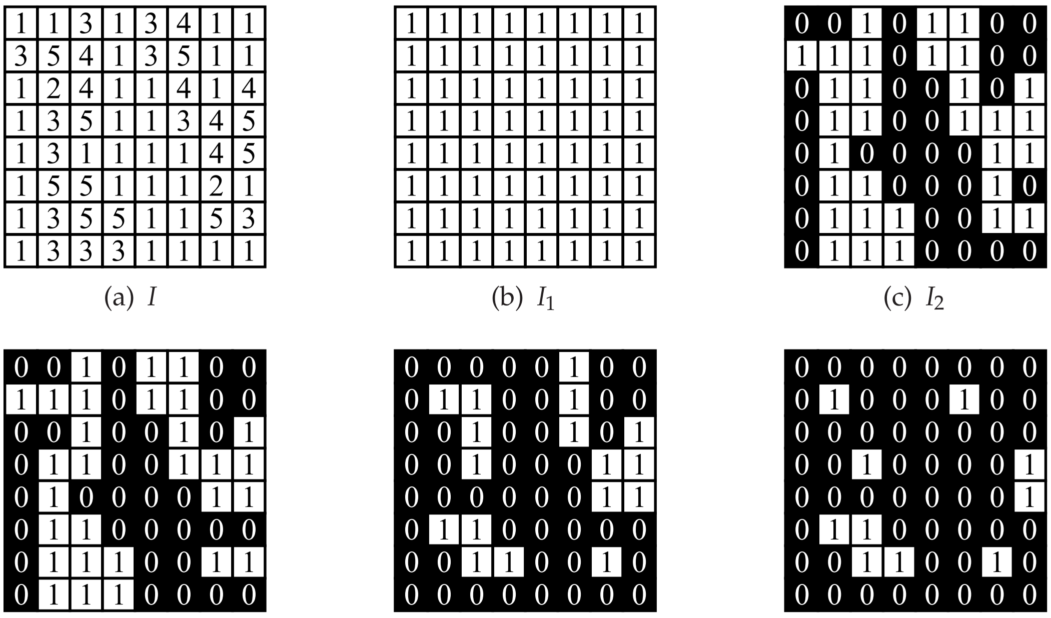
The *Component Tree* (CT) (Carvalho, Costa, Ferreira & Cesar-Jr., 2010) is a hierarchical representation of a digital image after thresholding operations between its minimum and maximum gray values (Mosorov & Kowalski, 2002). There exists a relation of inclusion between components at sequential gray levels in the image, explained below by the partition definition (Carvalho, 2004).

Definition 1. *A partition P of an image I is a set of disjoint regions R_i, $i \in \mathcal{N}$, where $\bigcup_i^n R_i = I$ and $R_i \cap R_j = \varnothing$, $i \neq j$.*

A cross-section I_k of an image I is a binary image defined as (Mosorov & Kowalski, 2002; Najman & Couprie, 2006):

$$I_k = \{x \in I / I(x) \geq k\}. \tag{2}$$

In the CT, the *Connected Components* (CC) of all cross-sections are organized in a tree structure. There exists an edge between two connected components CC^i_{k+1} and CC^j_k when $CC^i_{k+1} \subseteq CC^j_k$ (inclusion relation); k is a cross-section identifier, i is a CC in cross-section $k+1$, and j is a CC in cross-section k. The connected component of the first cross-section corresponds to the whole image domain and it is called root. The leaves are the elements of the CT that have no children. A fast algorithm to build the CT is given by Najman & Couprie (2006). Fig. 1 shows a gray scale image I and its five cross-sections I_k. The Component Tree for image I is depicted in Fig. 2(a).



1	1	3	1	3	4	1	1
3	5	4	1	3	5	1	1
1	2	4	1	1	4	1	4
1	3	5	1	1	3	4	5
1	3	1	1	1	1	4	5
1	5	5	1	1	1	2	1
1	3	5	5	1	1	5	3
1	3	3	3	1	1	1	1

(a) I

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

(b) I_1

0	0	1	0	1	1	0	0
1	1	1	0	1	1	0	0
0	1	1	0	0	1	0	1
0	1	1	0	0	1	1	1
0	1	0	0	0	0	1	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
0	1	1	1	0	0	0	0

(c) I_2

0	0	1	0	1	1	0	0
1	1	1	0	1	1	0	0
0	0	1	0	0	1	0	1
0	1	1	0	0	1	1	1
0	1	0	0	0	0	1	1
0	1	1	0	0	0	0	0
0	1	1	1	0	0	1	1
0	1	1	1	0	0	0	0

(d) I_3

0	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	0	0	0	1	1
0	0	0	0	0	0	1	1
0	1	1	0	0	0	0	0
0	0	1	1	0	0	1	0
0	0	0	0	0	0	0	0

(e) I_4

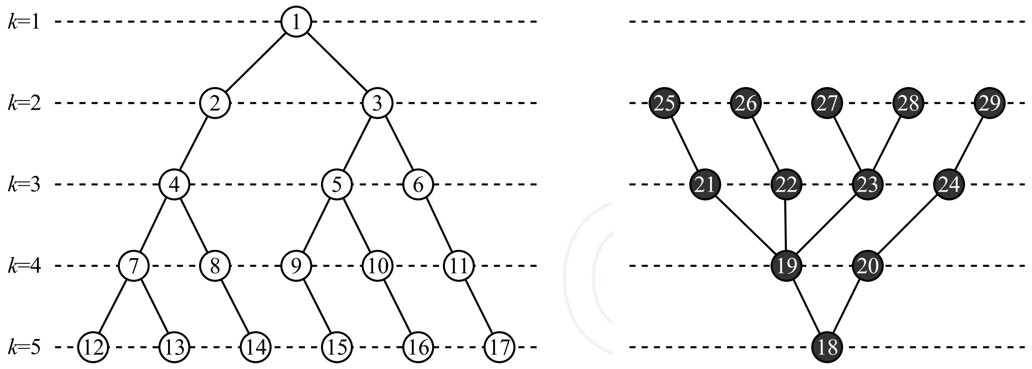
0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0
0	0	1	1	0	0	1	0
0	0	0	0	0	0	0	0

(f) I_5

Fig. 1. Thresholding image graylevels to build cross-sections. (a) a grayscale image I and (b)-(f) its five cross-sections.

When observing the traditional CT model, one realizes that the tree will be composed by only *white* components, *i. e.*, components with value 1. There still information in the cross-sections related to the *black* components. In fact, these components can provide more relevant information in particular cases. Therefore, we have defined the Reverse Component Tree (RCT) (Carvalho, Costa, Ferreira & Cesar-Jr., 2010) where two connected components CC^i_k and CC^j_{k+1} are linked when $CC^j_k \subseteq CC^i_{k+1}$; k is a cross-section identifier, i is a CC in cross-section $k+1$, and j is a CC in cross-section k. Unlike described for the CT case, the roots of the RCT's are formed by the connected components of the last cross-section. Fig. 2(b) shows the Reverse Component Tree for image I, presented in Fig. 1(a).

In order to build a connected similarity graph, we combine Component and Reverse Component Trees. This similarity graph will be used in the graph cut process. First, a connected subgraph $G_k = (V_k, E_k, W_k)$ is created for each cross-section I_k, where the nodes



(a) Component Tree

(b) Reverse Component Tree

Fig. 2. Component Tree and Reverse Component Tree from grayscale image presented in Fig. 1(a).

$v_{k,n}$ correspond to the connected components CC_k^n; $k \in \mathcal{N}$ is a cross-section identifier, and $n \in \mathcal{N}$ is a connected component identifier. Given two nodes $v_{k,i}$ and $v_{k,j}$, $v_{k,i} \sim v_{k,j}$ if distance$(v_{k,i}, v_{k,j}) \leq r$, where $r \in \mathcal{R}$ is the connection radius. After built, the k subgraphs are connected by adding the edges from the CT and RCT. The weight of each edge should reflect the likelihood of different connected components. Some of them are difference between areas, density, average gray levels and standard deviation, and Euclidean distance.

One strong characteristic of CT method is the generation of multiple image partitions at once. Because a cross-section I_k corresponds to the whole image, there is one image partition for each cross-section.

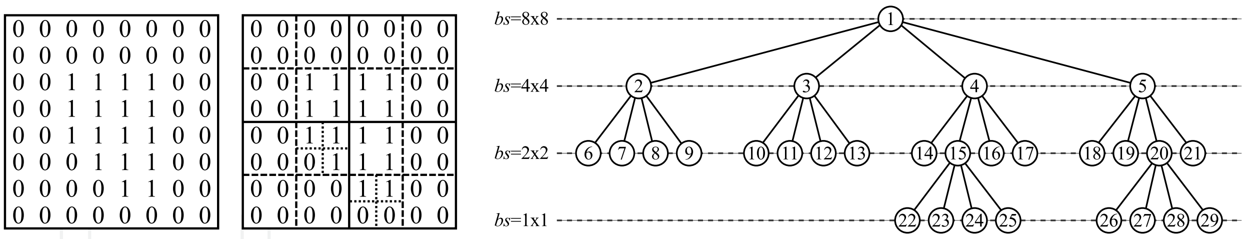
Finally, it is important to note that some images can produce a high quantity of connected components, especially in the presence of noise. Therefore, it is useful to apply some pre-processing on the image before starting the CT computation such as, gaussian filter, normalization of the gray values into a smaller range, and merging of identical subsequent cross-sections.

2.4 Graph based on the Quadtree

A Quadtree is a data structure formed from the recursive decomposition of a space (Samet, 1984). In the image processing domain, a quadtree usually maps an image and its regions into a directed acyclic graph (Consularo & Cesar-Jr., 2005).

The decomposition process is simple: The initial region corresponds to the whole image and is associated to the root node; each region in the image should be recursively decomposed into exact four new disjoint regions until they satisfy a defined criterion of homogeneity. Fig. 3 shows a Quadtree decomposition example. The decomposition criterion choice in this case was to exist regions with only one value.

In practice, the regularity of the Quadtree decomposition limits the application of this method to square images with edge sizes 2^n, $n \in \mathcal{N}$. One solution is to relax the regular decomposition, allowing a more suitable number of regions. But the greatest



(a) I (b) Regions (c) Quadtree

Fig. 3. Quadtree decomposition. (a) original image. (b) decomposed regions. (c) corresponding Quadtree.

difficulty in creating the quadtree is the choice of the decomposition criterion. There are different criteria proposed, such as the standard deviation or entropy of image gray levels (Consularo & Cesar-Jr., 2005).

In our quadtree approach (Carvalho, Costa & Ferreira, 2010) for image segmentation, we proposed applying the Canny (Canny, 1986) (1986) filter in the image before the decomposition. This filter has low sensitivity to noise in images. By removing pixels with low gradient and thresholding the resultant ones, this process results in a binary image with border pixels highlighted. This procedure was chosen because:

1. after the filtering, the image results in a binary matrix. Then, facilitating the decomposition criterion definition, that a region should be decomposed when it is not formed entirely by ones or zeros (Samet, 1984);
2. the edge detection operation drastically reduce the size of data to be processed, while at the same time preserves the structural information about object boundaries (Canny, 1986).

In our work (Carvalho, Costa & Ferreira, 2010), the main reason for using a Quadtree image representation was to reduce the similarity graph size. Thus, in this technique the graph is generated using only the regions associated to the Quadtree leaves. Each region is associated to a node and for each region a centroid pixel is defined as the representative pixel. The nodes of the similarity graph are linked together if their representative pixel distances are less than a given radius r, similar to the pixel grid approach. However, it is useful to consider the nodes region sizes to calculate the radius r.

The number of nodes on the similarity graph can be influenced by the choice of the edge detector parameters. The number of regions obtained by the proposed technique will vary according to the image features. Also, the parameters of the edge detection filter can be manually specified, in order to change its sensibility.

3. Graph Cut and Normalized Cut

A graph cut partitions the set of nodes V of a graph $G = (V, E, W)$ into two disjoint subsets A and B, and can be expressed by the following equation (Shi & Malik, 1997):

$$\text{Cut}(A, B) = \sum_{u \in A, v \in B} w(u, v), \tag{3}$$

where $w(u, v)$ are the edges removed from G.

This formula indicates the degree of dissimilarity between G_A and G_B, the corresponding subgraphs to the node subsets A and B, respectively. There are a lot of ways to solve the problem of graph cuts. One of them, proposed by Wu & Leahy (1993), solve this problem removing the smallest possible number of edges. However, in some cases this approach can produce isolated nodes. A study about types of graph cut is given by Soundararajan & Sarkar (2001).

A graph cut can be accomplished by means of *Spectral Graph Theory*. In this approach, the matrix eigenvectors that represent graphs are analyzed and used as parameter in order to partition a graph.

3.1 Spectral Graph Theory

The *Spectral Graph Theory* (SGT) studies the eigenvalues of the graph matrices, also called graph spectrum. Algebraic methods used to analyze matrices of graphs are especially effective in treating regular and symmetric graphs (Chung, 1997). The matrices commonly used are adjacency matrices and the SGT establishes a relation between the graph spectrum and the graph features.

The use of graph spectrum information for graph cuts has a great contribution from Donath & Hoffman (1972); Fiedler (1975); Pothen et al. (1990). Fiedler proposed that the second smallest eigenvector v_2 of the Laplacian matrix, also called the Fiedler vector, has in a given row $v_2[i]$ a numerical information about node i, also called the characteristic value of the node. The graph nodes can be partitionated by grouping them according to their value in the Fiedler vector. The commonly way used to group nodes is the characteristic values signals.

3.2 The Normalized Cut

The *Normalized Cut* (NCut) technique (Shi & Malik, 1997) is a theoretic method for graph partitioning. Its goal is to find a balanced cut in a graph, in order to generate two or more subgraphs. This technique solves the problem stated by Wu & Leahy (1993) in their minimum cut criteria for graph cutting. Applying this method for image segmentation is possible with a proper image-graph representation, where the subgraphs obtained from graph partitioning represents the image regions. The NCut in a graph G is calculated by the following equation:

$$\text{NCut}(A, B) = \frac{\text{Cut}(A, B)}{\text{SumCon}(A, V)} + \frac{\text{Cut}(A, B)}{\text{SumCon}(B, V)}, \tag{4}$$

where A and B are the node subsets of subgraphs G_A and G_B, subject to $A \cup B = V$ and $A \cap B = \varnothing$; $\text{Cut}(A, B)$ is defined in Equation (3); $\text{SumCon}(A, V)$ is the total weight of the edges connecting nodes from a subgraph G_A to all nodes in the original graph G; and $\text{SumCon}(B, V)$ is similarly defined to a subgraph G_B.

The optimal NCut is the one that minimizes Equation 4. The problem in minimizing Equation 4 is that it is only trivial for small graphs. For bigger graphs, it has a NP-Complete complexity. Shi & Malik (2000) extended this equation and found a well-known equation in linear algebra called the Rayleigh Quotient. It can be minimized using spectral graph properties of the graph's Laplacian Matrix described by Fiedler (1975), *i.e.*, its minimum value is λ_2, the second smallest eigenvalue (Golub & Loan, 1989).

The graph partitioning is guided by the eigenvector v_2, where each value $v_2[i]$ will represent a graph node i. To split a graph, a threshold value is used and the graph nodes are partitioned in two subsets. The most common threshold values are zero, the median value in v_2 or the one that minimizes the NCut value.

3.3 Similarity Graph

In a *Similarity Graph* the edge weights represent the degree of similarity between the linked nodes. For graphs that represent images, the similarity can be determined by a function of intensity, position and other image pixels features. A measure of similarity regarding the intensity and the position of image pixels is given by (Shi & Malik, 2000):

$$\mathrm{W_{IP}}(i,j) = \begin{cases} e^{-\left(\frac{\alpha^2}{d_p}\right)-\left(\frac{\beta^2}{d_i}\right)}, \text{ if } \alpha_2 < r \\ \qquad\qquad 0, \text{ Otherwise} \end{cases}, \tag{5}$$

where $\alpha = ||P_i - P_j||$ and $\beta = ||I_i - I_j||$ are respectively the distance and the difference of intensity between pixels i and j; r is a given distance (also called graph connection radius); and d_p and d_i are set as the variance of the image pixels positions and intensity. This grouping cue used separately often gives bad segmentations because some natural images are affected by the texture clutter.

The intervening contours is another measure to evaluate the affinity between two nodes by measuring the image edges between their correspondent pixels. The intervening contour similarity function is given by (Cour et al., 2005):

$$\mathrm{W_C}(i,j) = \begin{cases} e^{-\left(\frac{\max_{(x \in \text{ line}(i,j))} \varepsilon^2}{d_c}\right)}, \text{ if } \alpha < r \\ \qquad\qquad 0, \text{ Otherwise} \end{cases}, \tag{6}$$

where $\text{line}(i,j)$ is a straight line joining pixels i and j and $\varepsilon = ||\,\text{Edge}(x)||$ is the image edge strength at location x.

These two grouping cues can be combined as (Cour et al., 2005):

$$\mathrm{W_{IPC}}(i,j) = \sqrt{\mathrm{W_{IP}}(i,j)\,\mathrm{W_C}(i,j)} + \mathrm{W_C}(i,j). \tag{7}$$

3.4 Related work

There is a wide range of recent work in image segmentation using the Normalized Cut technique. The contributions are focused on improving the algorithm performance, others on proposing different image-graph modelling and others on the application of this technique for real-world applications.

In several works, the image model used is the similarity graph built by taking each image pixel as a node. In this case, the node pairs within a given radius r are connected by an edge. This graph will be explained in the next section.

Monteiro & Campilho (2008) proposed the Watershed Normalized Cut, which uses the regions from the Watershed image segmentation as nodes for the similarity graph. The Watershed region similarity graph is either used by Carvalho et al. (2009) for comparison with the primitive pixel affinity graph in yeast cells images segmentation. Ma & Wan (2008) used Watershed based similarity graphs to segment texture images.

The primitive normalized cut enhancement was also studied and applied by many researchers. Cour et al. (2005) proposes a Normalized Cut adaptive that focus on the computational problem created by long range graphs. The authors suggested the use of multiscale segmentations, decomposing a long range graph into independent subgraphs. The main contribution of this technique is that larger images can be better segmented with a linear complexity. Sun & He (2009) proposed the use of the multiscale graph decomposition, partitioning the image graph representation at the finest scale level and weighting the graph nodes using the texture features.

Tolliver & Miller (2006) suggested an improvement of the normalized cut technique. They proposed the use of the k first eigenvectors for graph partitioning as the k-way Normalized cut. The difference is that these eigenvectors modify the edges weight in the graph, resulting in new graph matrices, and the k first eigenvectors are calculated again in the new Laplacian Matrix. The authors proved that this procedure changes the k first eigenvalues to zero. Spectral graph theory concepts about the Laplacian matrix informs that the number of eigenvalues equal to zero shows the number of connected components in a graph. Their algorithm returns these connected components. Cour et al. (2005) suggested another improvement by dividing the graph in scales and processing them in paralell. This approach can segment large images graphs with high conections with linear complexity.

Tao et al. (2008) proposed a new image thresholding technique using the normalized cut. The graph similarity matrix proposed is now based on pixel gray levels, reducing the matrix size and the computational cost. So, a new matrix M is created, where $M(i,j) = \text{Cut}(V_i, V_j)$ with i and j being two given gray levels. Using this matrix, the normalized cut is then calculated to each threshold value, If the normalized cut related to a given threshold value t is below a prespecified value, this threshold value is adequate to separate the objects from the background in this image.

Grote et al. (2007) suggested the normalized cut for extracting roads from aerial images. In their graph, pixels are the graph nodes and the similarity matrix uses contours, hue and color in the image pixels. this approach uses the k-way normalized cut, with k being large to avoid road and non-road pixels mixture. Senthilnath & Omkar (2009) compared this technique with other state-of-art road extraction approach, proving that the normalized cut based technique works better. Other normalized cut aplications in the literature are the noise reduction in images by Zhang & Zhang (2009) and the colour image segmentation by Tao et al. (2008).

4. Normalized Cut segmentation workflow

Given an image I, we build a similarity graph $G = (V, E, W)$, from Quadtree and/or Component Tree representations. The image segmentation process based on Normalized Cut technique can be applied by two distinct methods: recursive Two-way and K-Way. The block diagram presented in Fig. 4 illustrates the complete image segmentation process.

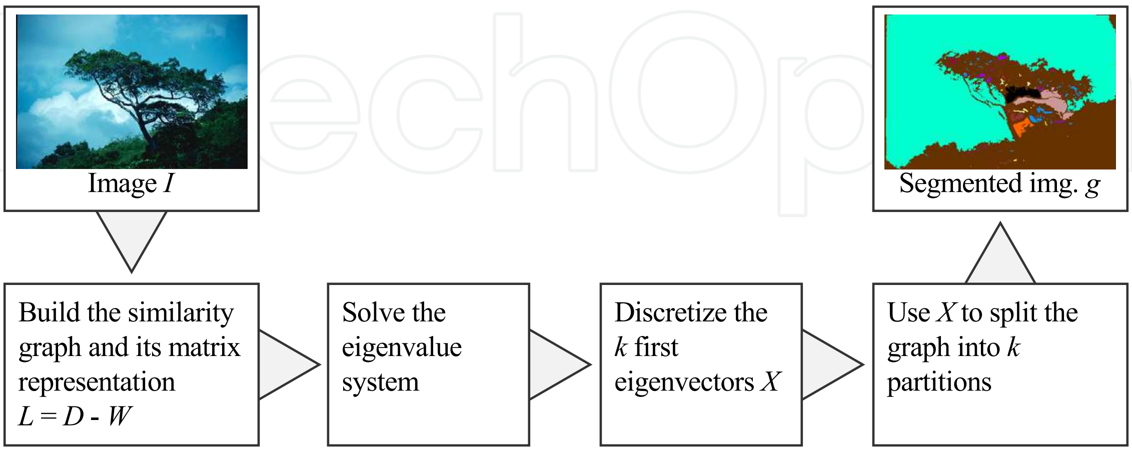


Image I
Segmented img. g
Build the similarity graph and its matrix representation $L = D - W$
Solve the eigenvalue system
Discretize the k first eigenvectors X
Use X to split the graph into k partitions

Fig. 4. Workflow of the image segmentation technique based on Normalized Cut.

The similarity graph obtained from Quadtree and Component/Reverse Tree representations is done according to the details explained previously and its parameters are described in Section 5. They have hierarchical information that is exploited in the graph cut process. The eigenvectors of the Laplacian matrix are obtained from the solution of the following equation:

$$(D - W)v = \lambda Dv. \tag{8}$$

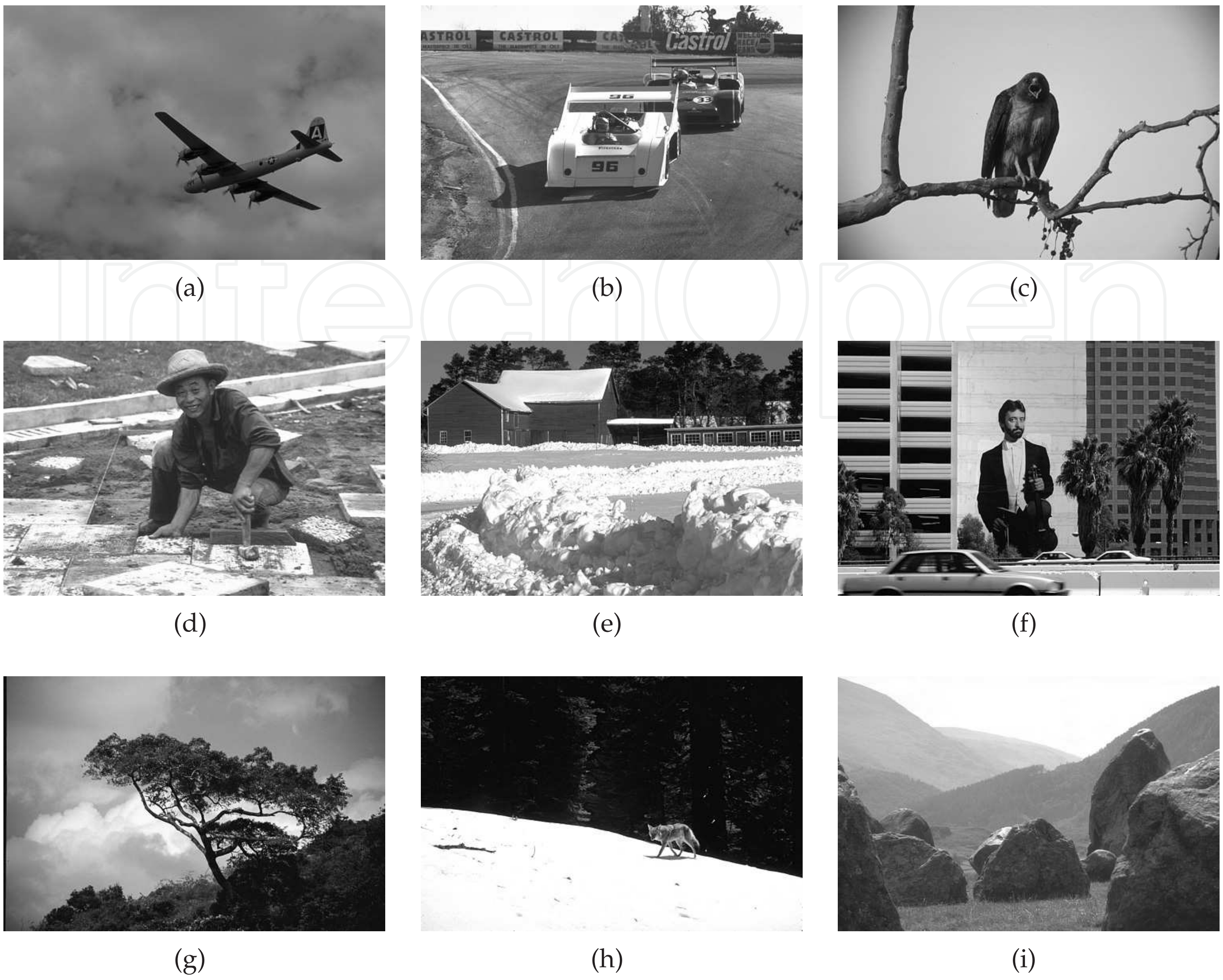
At this moment, it is possible to use only the second eigenvector v_2 in order to provide two partitions of the graph G and reapply the process to obtain recursively a large number of partitions. A partition should be divided by analysing a specified cut value. This technique is known as Recursive Two-Way NCut. In the other hand, it is possible to discretize the k first eigenvector X, were $X = [v1, v2, \dots, vk]$ and use them directly to implement the graph partitioning into k desired partitions. This partitioning process is called K-Way Cut and corresponds to the block sequence presented in Fig. 4.

5. Experimental results and discussion

We have performed k-way Normalized Cut segmentation in 100 grayscale test images from *Berkeley Segmentation Benchmark*[1] (Martin et al., 2001). The goal of these experiments is to compare the techniques for building the similarity graph based on *Pixel Grid* (Shi & Malik, 2000), *Multiscale Pixel Grid* (Cour et al., 2005), *Quadtree* (Carvalho, Costa & Ferreira, 2010), and *Component Tree* (Carvalho, Costa, Ferreira & Cesar-Jr., 2010). Fig. 5 show a collection of 9 selected images.

The Berkeley's benchmark rely on human image segmentations to state the segmentation algorithms assertiveness, according to Precision (P) and Recall (R) metrics (Davis & Goadrich,

[1] Available at http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ (last accessed September, 2011).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 5. Selected images from Berkeley's benchmark. (a) 3096. (b) 21077. (c) 42049. (d) 85048. (e) 97033. (f) 119082; (g) 147091. (h) 167062. (i) 241004.

2006; Martin et al., 2001). Precision is the probability that a pixel marked as a border is in fact a border pixel, and is given by

$$P = \frac{TP}{TP + FP}, \tag{9}$$

where TP is the number of true positives, and FP the number of false positives. The precision P decrease as increases the number of false positives. Recall, also called *hit rate*, is the probability that the border pixels marked by the machine are the same as the border pixels marked by humans, and is defined as

$$R = \frac{TP}{TP + FN}, \tag{10}$$

where FN is the number of false negatives.

These two metrics are summarized in the F-measure (Davis & Goadrich, 2006)

$$F = 2 \cdot \frac{P \cdot R}{P + R}, \tag{11}$$

that is used as the score metric by Berkeley's benchmark to ranking the algorithms effectiveness.

5.1 Experiment setup

The experiments were perfomed according to the workflow presented in Section 4 with k-way method. The connection radius for the *Pixel Grid* graph was $r = 10$ and the edges weights were given by Equation (6). For the *Multiscale Pixel Grid* approach were used one radius for each scale, which were $r_1 = 2$, $r_2 = 3$ and $r_3 = 7$ and the edge weights were given by Equation (6). The *Quadtree* weights were also given by Equation (6), and their radii given as

$$r_{i,j} = \frac{1}{2}\max(v_i^{\phi x} + v_j^{\phi x}, v_i^{\phi y} + v_j^{\phi y}) + k, \tag{12}$$

where $v_i^{\phi x}$ and $v_i^{\phi y}$ are repectively the horizontal and the vertical diameter of region correspondent to node $v_i \in V$; $v_j^{\phi x}$ and $v_j^{\phi y}$ are similarly defined for node $v_j \in V$; and $k = 10$.

The Component Tree was generated with radius $r = 25$ for the subgraphs. The attributes used to build our CT similarity graph were difference of area, distance, standard deviation of the gray levels and density. For the edges that links the subgraphs, the weights were multiplied by a factor given by the following equation:

$$f = \frac{NC_{Fi} + NC_{Fj}}{2 + |\,\mathrm{d}(i) - \mathrm{d}(j)|}, \tag{13}$$

where NC_{Fi} and NC_{Fj} are the number of CCs of the cross-sections that has the nodes i and j respectively; and $\mathrm{d}(i)$ and $\mathrm{d}(j)$ are the degrees of nodes i and j respectively, related to the CT. After the CT segmentation procedure the various partitions generated for each image were converted to a single one by two distinct approaches: manually by selecting the partition that seemed to be the better image segmentation by a criterion of resemblance to the original image; automatically by merging pairs of partitions with highest mutual information iteratively. These approaches sets our experiments on image segmentation based on CT as semi-automatic or automatic procedures.

5.2 Results, discussion, and future works

The Berkeley's benchmark combine the individual scores from all segmentations of each algorithm in a single final score that determine the algorithm overall ranking. In our experiments the first place in the ranking was for the *Multiscale Pixel Grid* approach and the last place was for the automatic *Component Tree*. Fig. 6 show the overall scores obtained for each similarity graph method.

The segmentation scores for some individual images, however, are quite different from the algorithm's overall score. Fig. 7 show the individual scores for each image with each algorithm. We can observe that even the automatic CT approach, which is the worst ranked algorithm, has the highest score to about ten images. A similar result was obtained by the Semi-automatic CT approach. Put together, they account for 20% of the better scores for individual images. Not surprisingly, they also have the highest overall *Recall*, 0.65 and 0.64

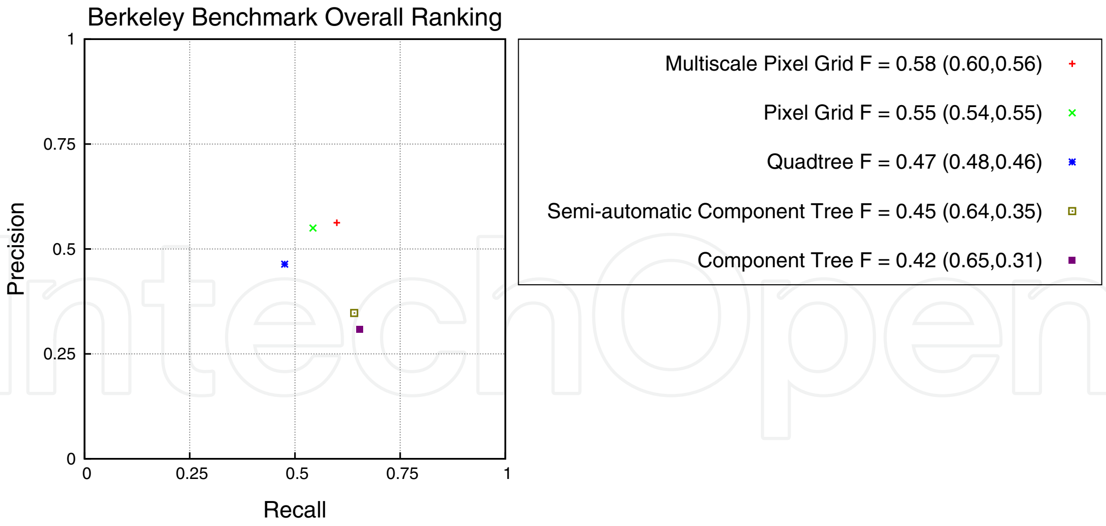


Fig. 6. Algorithms overall ranking. The two values between parenthesis at the right box represent (R,P).

for automatic and semi- automatic methods, respectively, against 0.60 for the best ranked algorithm.

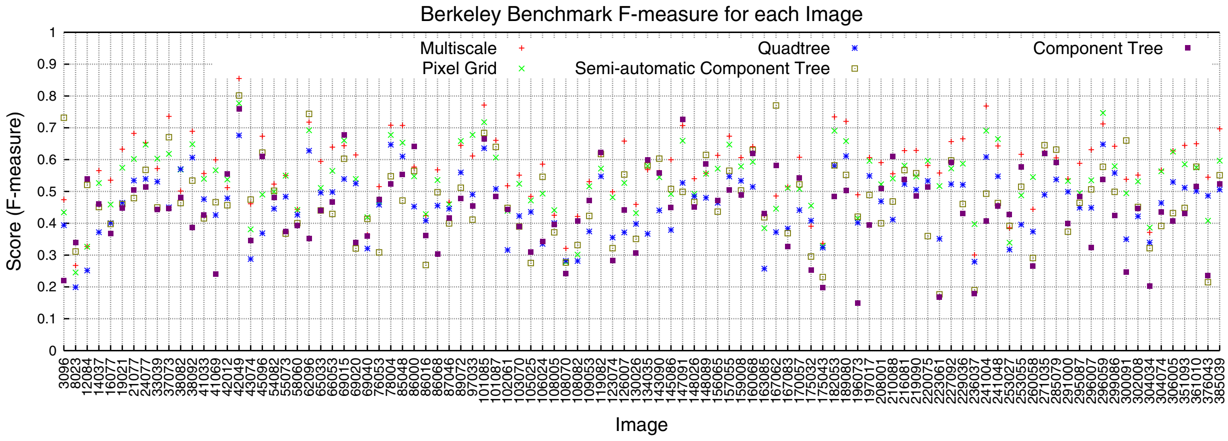


Fig. 7. Berkeley benchmark score (F-measure) for each image.

In the chart shown at Fig. 7 one can yet observe a clear trend for the Multiscale and the Quadtree approaches to follow the scores obtained by the classical Pixel Grid method: Multiscale with a little better, and Quadtree with a little worse scores. We can associate this trend to the fact that these algorithms are using the same similarity function.

Despite these quantitative analisys, it is also important to make a qualitative analisys of the segmentation results. Thus, are shown in Fig 8 one machine segmentation for each image shown in Fig. 5, and its human segmentations.

The computational performance is also an important requirement for the Normalized Cut technique. When Cour et al. (2005) proposed the Multiscale approach to generate the similarity graph, one main objective was to reduce the NCut computational cost. It is also the objective of the Quadtree approach. Fig. 9 show a chart with the time each algorithm took to process the segmentation. There is a strong correlation between the Component Tree

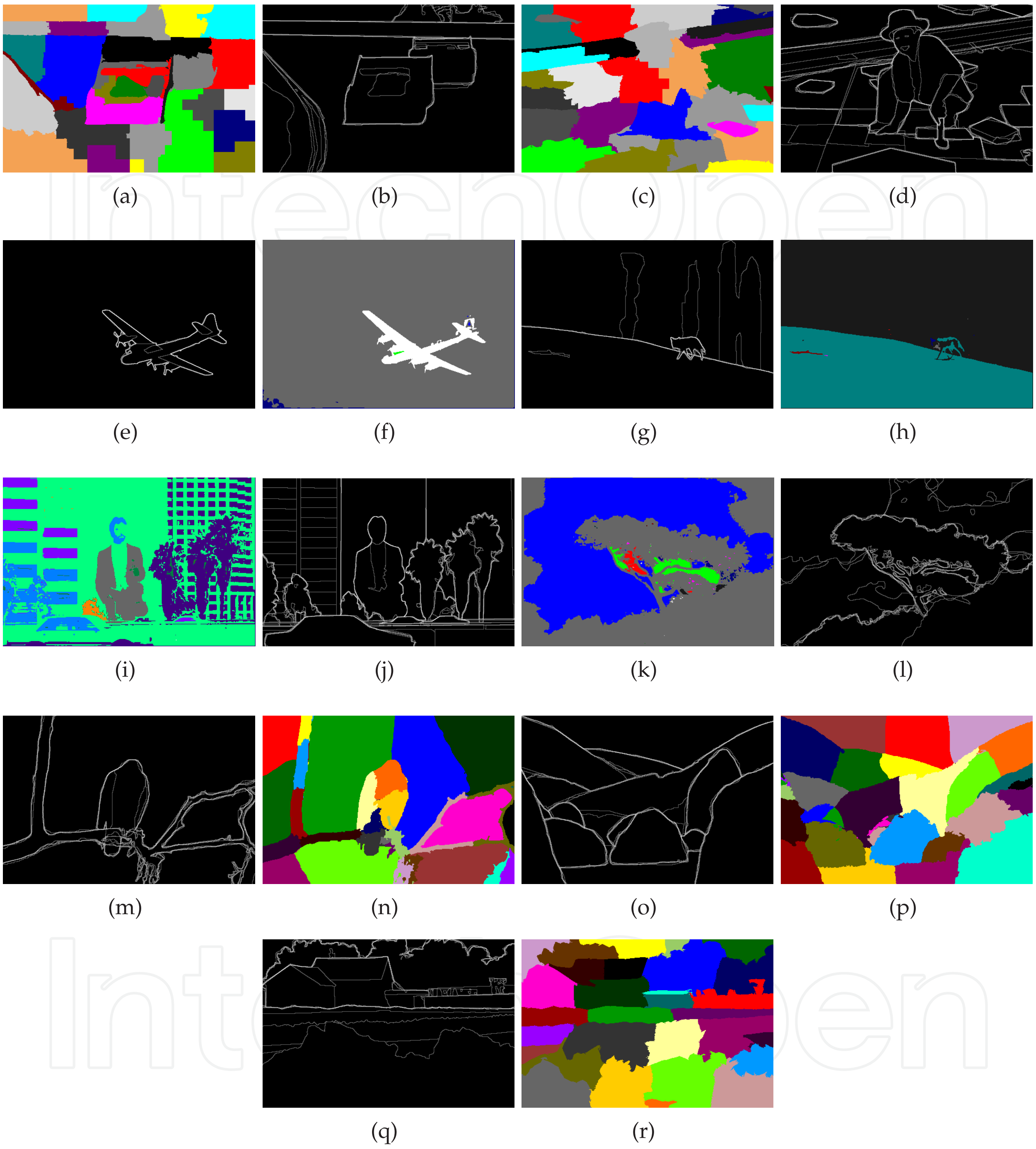
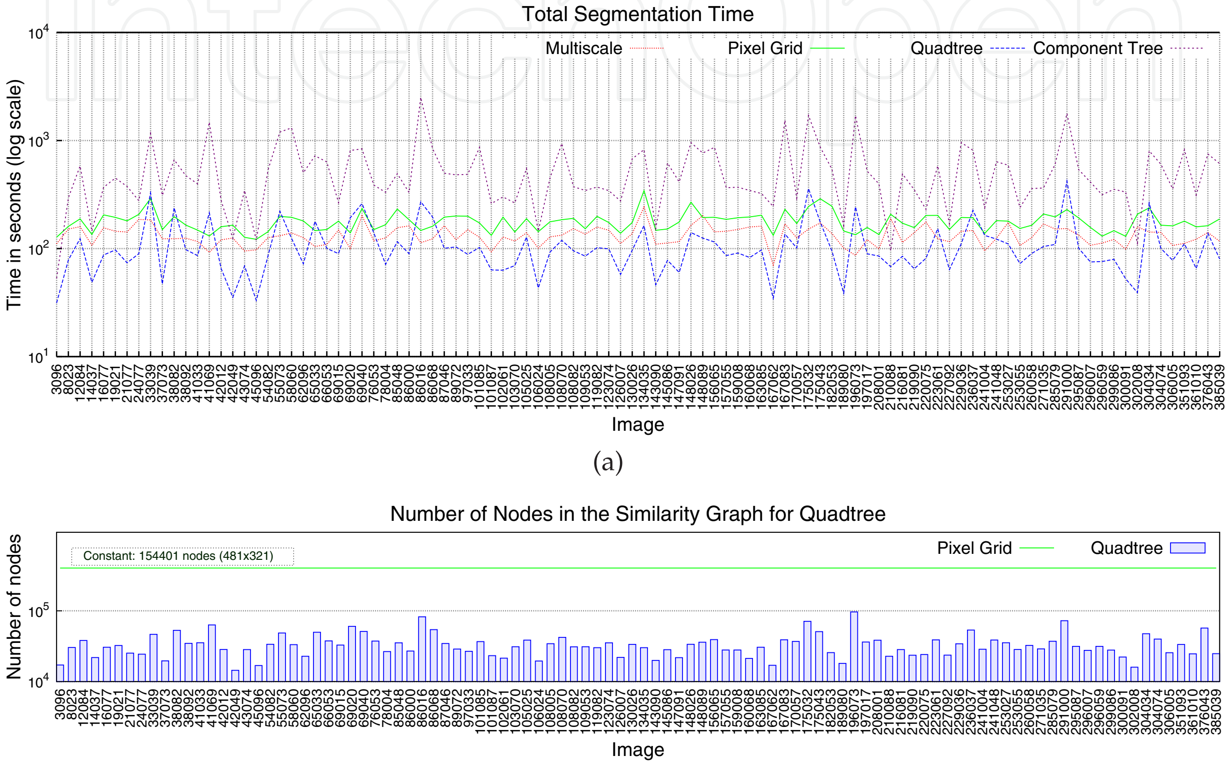


Fig. 8. Selected segmentations. (a) 21077 and (c) 85048 segmentations by Quadtree method. (f) 3096 and (h) 167062 segmentations by Semi-automatic Component Tree method. (i) 119082 and (k) 147091 segmentations by Component Tree method. (n) 42049 and (p) 241004 segmentations by Multiscale Pixel Grid method. (r) 97033 segmentation by Pixel Grid method. (b) 21077, (d) 85048, (e) 3096, (g) 167062, (j) 119082, (l) 147091, (m) 42049, (o) 241004, and (q) 97033, human segmentations from Berkeley benchmark.

and the Quadtree methods. That indicates that they are similarly sensitive to the image's data, but their overall performance are, respectively, the lower and the higher ones. The overall performance of the Multiscale algorithm is higher than the overall performance of the Pixel Grid algorithm, but is lower than the performance of the Quadtree method. Despite the lower performance, only the Component Tree can generate multiple image partitions at once. This problem can be alleviated by implementing the Najman's (Najman & Couprie, 2006) fast algorithm to build the Component Tree.



(a)

(b)

Fig. 9. Algorithms performance. (a) execution time for each image. (b) number of nodes in the similarity graph for Quadtree representation.

The chart in Fig. 9(b) show the number of nodes in the similarity graph generated by the Quadtree approach. Notice that the number of regions in the graph has direct impact to the algorithm's performance.

For future works we see that the Component Tree is the most promising method, despite its worst effectiveness and efficiency. Its implementation can be improved to reduce the computational cost. A more detailed study about the similarity criteria has the potencial of reducing the false positive rate.

There are future works for the Quadtree approach as well. Once the nodes represent regions instead of pixels, the study of texture and other region-based similarity criteria would improve the method effectiveness. It is also reasonable to explore further the hierarchical information of the Quadtree, that would lead the design of a new multiscale approach.

6. Conclusion

In this Chapter we proposed an approach to implement image segmentation based on graph modelling and Normalized Cut technique. Performing graph partitioning by means of Normalized Cut has been widely used in the specific literature. The possibility of generate balanced partitions has shown that this approach is efficient. The proposed similarity graphs, build from the Quadtree and Component Tree structures, proved promising compared to the traditional modelling based on pixel similarity graph. We performed comparisons using two-well established metrics, the Precision and Recall values. An additional aspect to be considered in the proposed graph models is exploring the hierarchical structures of both, Quadtree and Component Tree. Besides, we exploit a little more the Component Tree and proposed the Reverse Component Tree as a way to better represent the information contained in the image cross-sections.

The experimental results accomplished on images from the Berkeley Database show that use regions, or primitive regions to be specific, instead pixels seems to be a better strategy to segment images by Normalized Cut approach. In addition, the new image representation had the advantage of reducing the number of graph nodes and, therefore, improved the algorithm performance.

7. Acknowledgment

This work is supported by FAPESP (São Paulo Research Foundation) – Proc. 2010/14759-0.

8. References

Canny, J. (1986). A computational approach to edge-detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 679–700.

Carvalho, M. A. G. (2004). *Análise hierárquica de imagens através da árvore dos lagos críticos*, PhD thesis, Faculdade de Engenharia Elétrica e Computação- FEEC.

Carvalho, M. A. G., Costa, A. L. & Ferreira, A. C. B. (2010). Image segmentation using quadtree-based similarity graph and normalized cuts, Lecture Notes in Computer Science, v. 6419, pp. 329-337.

Carvalho, M. A. G., Costa, A. L., Ferreira, A. C. B. & Cesar-Jr., R. M. (2010). Image segmentation using component tree and normalized cuts, *Proceedings of the XXIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI2010)*, Gramado - Brazil, pp. 317-320.

Carvalho, M. A. G., Ferreira, A. C. B., Pinto, T. W. & Cesar-Jr., R. M. (2009). Image segmentation using watershed and normalized cuts, *Proc. of 22th Conference on Graphics, Patterns and Images (SIBGRAPI)*, Rio de Janeiro - Brazil.

Chung, F. (1997). *Spectral Graph Theory*, American Mathematical Society.

Consularo, L. A. & Cesar-Jr., R. M. (2005). Quadtree-based inexact graph matching for image analysis, *Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI2005)*, Natal - Brazil, pp. 205–212.

Cour, T., Bénézit, F. & Shi, J. (2005). Spectral segmentation with multiscale graph decomposition, *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR'05*, Vol. 2, pp. 1124–1131.

Davis, J. & Goadrich, M. (2006). The relationship between precision-recall and roc curves, *ICML '06: Proceedings of the 23rd international conference on Machine learning*, ACM, pp. 233–240.

Donath, W. & Hoffman, A. (1972). Algorithms for partitioning graphs and computer logic based on eigenvectors of connection matrices, *Technical report*, IBM.

Fiedler, M. (1975). A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory, *Czech. Math. Journal* 25(100): 619–633.

Golub, G. H. & Loan, C. F. V. (1989). *Matrix Computations*, John Hopkins Press.

Gonzales, R. & Woods, R. (2000). *Digital Image Processing*, Addison-Wesley.

Grote, A. et al. (2007). Segmentation based on normalized cuts for the detection of suburban roads in aerial imagery, *IEEE Proceedings of Urban Remote Sensing Joint Event* pp. 1–5.

Ma, X. & Wan, W. (2008). Texture image segmentation on improved watershed and multiway spectral clustering, *Proceedings of International Conference on Audio, Language and Image Processing - ICALIP*, pp. 1693–1697.

Martin, D., Fowlkes, C., Tal, D. & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, *Proc. 8th Int'l Conf. Computer Vision*, Vol. 2, pp. 416–423.

Monteiro, F. C. & Campilho, A. (2008). Watershed framework to region-based image segmentation, *Proc. of IEEE 19th International Conference on Pattern Recognition - ICPR*, pp. 1–4.

Mosorov, V. & Kowalski, T. M. (2002). The development of component tree for grayscale image segmentation, *Proc. of International Conference on Moderns Problems of Radio Engineering, Telecommunications and Computer Science – TECSET*, Slavsko – Ukraine, pp. 252–253.

Najman, L. & Couprie, M. (2006). Building the component tree in quasi-linear time, *IEEE Transactions on Image Processing* 15(11): 3531–3539.

Pothen, A., Simon, H. D. & Liou, K.-P. (1990). Partitioning sparse matrices with eigenvectors of graphs, *SIAM J. Matrix Anal. Appl.* 11: 430–452.
URL: *http://portal.acm.org/citation.cfm?id=84514.84521*

Samet, H. (1984). The quadtree and related hierarchical structures, *ACM Computing Surveys* 16(2): 187–261.

Senthilnath, M. R. & Omkar, S. N. (2009). Automatic road extraction using high resolution satellite image based on texture progressive analysis and normalized cut method, *Journal of the Indian Society of Remote Sensing* 37(3): 351–361.

Shi, J. & Malik, J. (1997). Normalized cuts and image segmentation, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 731–737.

Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 22(8) pp. 888-905.

Soundararajan, P. & Sarkar, S. (2001). Analysis of mincut, average cut and normalized cut measures, *Workshop on Perceptual Organization in Computer Vision*.

Spielman, D. (2007). Spectral graph theory and its applications, *Proc. of 48th Annual IEEE Symposium on Foundations of Computer Science*, pp. 29–38.

Sun, F. & He, J. P. (2009). A normalized cuts based image segmentation method, *Proc. of II International Conference on Information and Computer Science*, pp. 333–336.

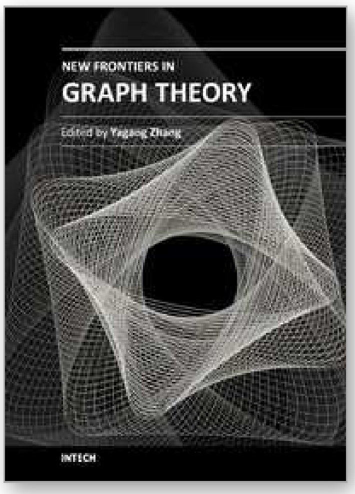
Tao, W., Jin, H., Zhang, Y., Liu, L. & Wang, D. (2008). Image thresholding using graph cuts, *IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans* 38(5): 1181–1195.

Tolliver, D. & Miller, G. (2006). Visual grouping and object recognition, *Graph partitioning by spectral rounding: Applications in image segmentation and clustering*, pp. 1053–1060.

Wilson, R. & Watkins, J. (1990). *Graphs: An Introductory Approach*, John Wiley and Sons.

Wu, Z. & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: theory and its application to image segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, pp. 1,101–1,113.

Zhang, L. & Zhang, J.-Z. (2009). Image denoising based on statistical jump regression analysis and local segmentation using normalized cuts, *Acoustics, Speech, and Signal Processing, IEEE International Conference on* pp. 661–664.



New Frontiers in Graph Theory
Edited by Dr. Yagang Zhang

ISBN 978-953-51-0115-4
Hard cover, 526 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012

Nowadays, graph theory is an important analysis tool in mathematics and computer science. Because of the inherent simplicity of graph theory, it can be used to model many different physical and abstract systems such as transportation and communication networks, models for business administration, political science, and psychology and so on. The purpose of this book is not only to present the latest state and development tendencies of graph theory, but to bring the reader far enough along the way to enable him to embark on the research problems of his own. Taking into account the large amount of knowledge about graph theory and practice presented in the book, it has two major parts: theoretical researches and applications. The book is also intended for both graduate and postgraduate students in fields such as mathematics, computer science, system sciences, biology, engineering, cybernetics, and social sciences, and as a reference for software professionals and practitioners.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821